

# UNIX - USER ADMINISTRATION

There are three types of accounts on a Unix system:

1. **Root account:** This is also called superuser and would have complete and unfettered control of the system. A superuser can run any commands without any restriction. This user should be assumed as a system administrator.
2. **System accounts:** System accounts are those needed for the operation of system-specific components for example mail accounts and the sshd accounts. These accounts are usually needed for some specific function on your system, and any modifications to them could adversely affect the system.
3. **User accounts:** User accounts provide interactive access to the system for users and groups of users. General users are typically assigned to these accounts and usually have limited access to critical system files and directories.

Unix supports a concept of *Group Account* which logically groups a number of accounts. Every account would be a part of any group account. Unix groups plays important role in handling file permissions and process management.

## Managing Users and Groups:

There are three main user administration files:

1. **/etc/passwd:** Keeps user account and password information. This file holds the majority of information about accounts on the Unix system.
2. **/etc/shadow:** Holds the encrypted password of the corresponding account. Not all the system support this file.
3. **/etc/group:** This file contains the group information for each account.
4. **/etc/gshadow:** This file contains secure group account information.

Check all the above files using **cat** command.

Following are commands available on the majority of Unix systems to create and manage accounts and groups:

Command	Description
useradd	Adds accounts to the system.
usermod	Modifies account attributes.
userdel	Deletes accounts from the system.
groupadd	Adds groups to the system.
groupmod	Modifies group attributes.
groupdel	Removes groups from the system.

You can use [Manpage Help](#) to check complete syntax for each command mentioned here.

## Create a Group

You would need to create groups before creating any account otherwise you would have to use existing groups at your system. You would have all the groups listed in */etc/groups* file.

All the default groups would be system account specific groups and it is not recommended to use them for ordinary accounts. So following is the syntax to create a new group account:

```
groupadd [-g gid [-o]] [-r] [-f] groupname
```

Here is the detail of the parameters:

Option	Description
-g GID	The numerical value of the group's ID.
-o	This option permits to add group with non-unique GID
-r	This flag instructs groupadd to add a system account
-f	This option causes to just exit with success status if the specified group already exists. With -g, if specified GID already exists, other (unique) GID is chosen
groupname	Actual group name to be created.

If you do not specify any parameter then system would use default values.

Following example would create *developers* group with default values, which is very much acceptable for most of the administrators.

```
$ groupadd developers
```

## Modify a Group:

To modify a group, use the **groupmod** syntax:

```
$ groupmod -n new_modified_group_name old_group_name
```

To change the *developers\_2* group name to *developer*, type:

```
$ groupmod -n developer developer_2
```

Here is how you would change the financial GID to 545:

```
$ groupmod -g 545 developer
```

## Delete a Group:

To delete an existing group, all you need are the **groupdel** command and the group name. To delete the financial group, the command is:

```
$ groupdel developer
```

This removes only the group, not any files associated with that group. The files are still accessible by their owners.

## Create an Account

Let us see how to create a new account on your Unix system. Following is the syntax to create a user's account:

```
useradd -d homedir -g groupname -m -s shell -u userid accountname
```

Here is the detail of the parameters:

Option	Description
-d homedir	Specifies home directory for the account.
-g groupname	Specifies a group account for this account.
-m	Creates the home directory if it doesn't exist.
-s shell	Specifies the default shell for this account.
-u userid	You can specify a user id for this account.
accountname	Actual account name to be created

If you do not specify any parameter then system would use default values. The `useradd` command modifies the `/etc/passwd`, `/etc/shadow`, and `/etc/group` files and creates a home directory.

Following is the example which would create an account *mcmohd* setting its home directory to */home/mcmohd* and group as *developers*. This user would have Korn Shell assigned to it.

```
$ useradd -d /home/mcmohd -g developers -s /bin/ksh mcmohd
```

Before issuing above command, make sure you already have *developers* group created using *groupadd* command.

Once an account is created you can set its password using the **passwd** command as follows:

```
$ passwd mcmohd20
Changing password for user mcmohd20.
New UNIX password:
Retype new UNIX password:
passwd: all authentication tokens updated successfully.
```

When you type *passwd accountname*, it gives you option to change the password provided you are super user otherwise you would be able to change just your password using the same command but without specifying your account name.

## Modify an Account:

The **usermod** command enables you to make changes to an existing account from the command line. It uses the same arguments as the `useradd` command, plus the `-l` argument, which allows you to change the account name.

For example, to change the account name *mcmohd* to *mcmohd20* and to change home directory accordingly, you would need to issue following command:

```
$ usermod -d /home/mcmohd20 -m -l mcmohd mcmohd20
```

## Delete an Account:

The **userdel** command can be used to delete an existing user. This is a very dangerous command if not used with caution.

There is only one argument or option available for the command: `-r`, for removing the account's home directory and mail file.

For example, to remove account *mcmohd20*, you would need to issue following command:

```
$ userdel -r mcmohd20
```

If you want to keep her home directory for backup purposes, omit the `-r` option. You can remove the home directory as needed at a later time.